1. **Fitt's Law**

   In our design, we used Fitts's Law to intentionally choose the size and placement of our buttons. For example, when a user is signed in, we want to make it very clear where they can sign out, so if one goes to the sign in page, the only button that is displayed is a large blue button reading "Sign Out" in very large letters near the top of the page, so that the user's eyes are immediately drawn and they can easily see what to click in order to sign out. For deleting one's account, we used a very large red button that reads "Delete User" in large letters so that the user is aware that this is a very permanent and significant action, however, we placed it at the bottom of the page, as we do not want a user's eyes to directly go to Delete User because this is an action that happens fairly infrequently.

2. **Speak the User's Language**

   We display error messages either at the top of the page or near the button of the action that the user is trying to perform that are easily interpretable to the user if they try to do an action that is either a bad request, or is not authenticated. For example, if a user tries to create an account with a name that is already taken, near the "Change Username" button, an error message reading "Username {attempted username} already exists." We do not return the status code to the user, and we do not merely return the JSON, as these are not easily interpretable to them.

3. **Consistent Naming & Icons**

   Throughout the UI, we use consistent naming and icons so that the user has an easy time navigating. One such instance is that everything is named in terms of a freet. If one wants to view freets, there is a tab that reads "Freet feed". If they would like to post a freet, there is a box labeled "Create Freet". If they would like to repost someone else's freet, there is a "Refreet" button. Calling these short messages 'freets' throughout the UI makes it so that there is no ambiguity as to what these messages are. Simply calling freets 'messages' could be confusing, as 'messages' could imply many possible things. Additionally, we use the same icon for refreeting and upvoting in the feed for every single freet. There is the same button clearly marked 'Refreet' appearing on every single freet in the feed, and the upvote button for every single freet is a heart icon, which the user can easily infer is to like a freet.

4. **Information Scent**

   It is important that the user can easily navigate through the UI, no matter which page they are currently on. To do this, we created a tabs header at the top of the UI that lists the three possible pages for our Fritter app: the sign-in page, the feed page, and the user profile page. These tabs are named so that the user can easily infer which page to navigate to: "Sign In/Out", "Feed", or "Profile". Clicking on any of these three tabs will route the user to the corresponding page.

5. **Show Location & Structure**

Going along with #4 (information scent), when the user is on that page, the tab is a darker shade of gray than the rest of the header, the font is bolded, and there is a blue border on the bottom (to visually represent a tab being pressed). This is so that the user can easily know which page they are currently on at all times without needing to remember which page they navigated to most recently. They can easily navigate to any of the three tabs from the current tab that they are currently on.

6. **Keep Paths Short**
   We intentionally designed our UI so that a user does not need to navigate through nested paths and links to get to a certain page. We kept our navigation simple with 3 main tabs, and from these 3 main tabs a user can access everything they need to easily: the feed (where they can read, post, update, delete, upvote, or refreet freets as well as follow users), the sign in/out page (in which they can create an account, sign in, or log out), as well as the profile page (where they can see their own freets, see their followers/following, or change their username/password as well as delete their account).

7. **Gestalt Principle of Grouping**
   We grouped together forms and actions that are similar in nature on multiple levels. From the highest level, we grouped all actions related to freets themselves/other users in the feed page, all actions related to updating one's own account in the profile page, and all actions related to logging in/out on the sign in/out page. Within those pages, actions are also grouped together based on similarity. For example, on the profile page, the change username and change password forms appear next to each other, and these appear next to the user's username (so when they change their username, they see it updated immediately). Another example is that creating freets and viewing freets appear near each other in the feed page, as these actions both relate to freets. Additionally, freets are grouped together on the feed page, as it is likely that if one is viewing freets, they would like to see all of them in one place. Because of this, there is a clearly marked box on the feed page in which all freets appear, and the user can scroll through these freets.

8. **Anticipation & Context**
   We chose our specific placement of actions based on the context of the page, as well as what we anticipate the user would like to do. For example, if a user navigates to the Profile page, it is likely because they would like to update their account in some manner, so we placed the Change Username and Change Password buttons on this page. They were initially on the Sign In/Out page, but it is unlikely that a user would intentionally navigate to the Sign In page to change their username. Additionally, like previously mentioned, all actions for a freet appear on the Freet page, as a user is likely navigating to the freet page to either view freets or perform some action on a freet. Thus, if a user navigates to the Freet page, the actions Create Freet, List Freets (viewing), Edit Freet, Delete Freet, Refreet, and upvote will appear. We also chose to remove actions that a user cannot perform if they are not signed in. For example, a user can not change their own username if they are not signed into an account, so these buttons disappear.